IEE PROCEEDINGS

SOFTWARE

SEN Vol. 149 No. 5 October 2002

special issue on

EMPIRICAL SOFTWARE ENGINEERING

Publication acknowledged by the Convention of Societies of Engineers of Europe EUREL

Published by IEE in association with BCS, and in co-operation with AEI, MEE, SEP, SER, SEV/ASE, VDE, with the support of

BCS

IEE Proceedings

SOFTWARE

The Institution of Electrical Engineers, Savoy Place, London WC2R 0BL, United Kingdom
Publishing Department: Michael Faraday House, Six Hills Way, Stevenage, Herts. SG1 2AY, United Kingdom

October 2002 Volume 149 Number 5 UK ISSN 1462-5970 IPSEFU 149 (5) 113–166

PAPERS

Papers should be addressed to the Managing Editor, *IEE Proceedings-Software*, Publishing Department, Institution of Electrical Engineers, Michael Faraday House, Six Hills Way, Stevenage, Herts. SG1 2AY, United Kingdom; telephone: +44(0)1438 767356; email: proceedings@iee.org.uk; facsimile: +44(0)1438 742849. Notes for the guidance of authors may be found on the inside back cover. **Papers of outstanding merit are eligible for IEE premiums.**

SUBSCRIPTIONS

There are twelve *IEE Proceedings* titles. Six issues of each title are published each year.
Prices applicable outside the Americas/inside the Americas:
Annual subscription prices for 2002 (paper or online, including delivery):

1 title	£515.00/$900.00	5 titles	£1265.00/$2215.00	9 titles	£1560.00/$2720.00
2 titles	£815.00/$1420.00	6 titles	£1375.00/$2405.00	10 titles	£1600.00/$2795.00
3 titles	£980.00/$1710.00	7 titles	£1450.00/$2535.00	11 titles	£1640.00/$2870.00
4 titles	£1145.00/$2000.00	8 titles	£1510.00/$2635.00	12 titles	£1680.00/$2940.00

Each additional title subscription in excess of twelve titles: £140.00/$245.00

Single copy: £86.00/$150.00; Airmail supplement: £20.00/$35.00 per title

A combined subscription to the Online and Printed version of *IEE Proceedings* is charged at 1.5 times the listed price. Online Journals are subject to VAT where applicable.
Orders should be addressed to IEE Publications Sales Department, Michael Faraday House, PO Box 96, Stevenage, Herts. SG1 2SD, United Kingdom; telephone: +44 (0)1438 767264; facsimile: +44 (0)1438 742792; e-mail: sales@iee.org.uk

IEE PROCEEDINGS ONLINE

All twelve titles of *IEE Proceedings* are available online through the IEE Online Journals service on the IEE's World-Wide Web server. Articles are stored in PDF format and may be viewed and printed using the Adobe® Acrobat® Reader. Features include: browsable table of contents pages for each journal issue, cross-journal searching of bibliographic records (inc. authors, titles, abstracts), and the ability to print articles locally. The issues are added in advance of print publication. For further information, please look at http://ioj.iee.org.uk/ or contact the Marketing Department at Stevenage.

ADVERTISING

Inquiries should be addressed to the IEE Advertisement Department at the Stevenage address.

Chief Executive of the IEE: Alf Roberts BSc PhD CEng FIEE FInstP
Managing Editor: Stuart Govan; *Editorial Manager:* Shirley Rossall; *Editorial Assistant:* Lee Baldwin
Formerly published as *Software Engineering Journal*
A publication of the Institution of Electrical Engineers in association with the British Computer Society

Editorial

Special Issue: Empirical Software Engineering

One of the paradoxes of software engineering is that, although it extensively employs concepts and practices that are drawn from experience and observation (for example, Information Hiding, Design Patterns, etc.), we rarely possess any empirical validation of these ideas that could link theory and concepts to observed practices. Although the last decade has seen a growth in interest in adopting empirical techniques for use in software engineering, these are still far from being incorporated into mainstream practice, even for academic research projects. Also, while a few areas have been subjected to quite extensive empirical study (most notably that of Inspection Techniques), many key concepts and practices have had little attention (for example, the CMM 'quality' culture; the use of design patterns; the usability of UML).

The nature of software, and hence of software development, and in particular the complexity that this involves, make empirical studies difficult to perform. To date, this has not been seen as a challenge to the legitimacy of our discipline. Yet, if software engineering is to become an engineering discipline of any sort, an underpinning of empirical studies can only be regarded as absolutely essential.

There are two major issues that need to be addressed:

1. What techniques do we need for empirical software engineering?
2. What are the 'grand challenges' that empirical software engineering should be addressing as a matter of priority?

The first of these involves us in identifying the type of questions that empirical software engineering should be addressing, and then identifying the techniques that might be best employed for addressing them. (For example, how do we derive conclusions from observational data sets that are 'sparse' and possible 'skewed'?) The second question is more domain-focused, and involves the identification of those areas of software engineering where there are important questions that can only be answered by empirical study, or where empirical studies may help to identify new and better practices. Although the two questions are clearly related, they are also distinct, one being concerned with the more academic emphasis upon methodological issues, the second with the domain and with the needs of industry.

We attempt to address both these issues in our annual Evaluation and Assessment in Software Engineering (EASE) conference held at Keele University. In this Special Issue, we present a selection of papers from EASE 02 that address a number of practical and methodological concerns.

Michael Halling and Stefan Biffl present an investigation of the efficiency of inspection meetings. Their paper gives an indication of the experimental precision that can be used when investigating a relatively self-standing process such as document inspections. This can be contrasted with the investigation of test-first programming performed by Matthias Müller and Oliver Hagner. They investigated testing from the viewpoint of extreme programming (XP). XP involves far more than test-first programming but to make any headway with empirical studies of a complex software engineering methods, it is necessary to study different parts of the process in isolation. However, this runs the risks that the results may not be applicable when the process component is used as part of the full process.

Case studies of actual software engineering practices are an important methodological tool for assessing software engineering techniques. Two papers in this Special Issue use a case study approach. Daniel Karlstöm, Per Runeson and Claes Wohlin investigated a method for determining process improvement strategies in an industrial setting in Japan. Martin Höst, Enrico Johansson, Adam Noren and Lars Bratthall investigated a benchmarking method that involved facilitating co-operation between two software companies.

While case studies allow us to investigate some phenomena in detail in a realistic setting, surveys give a wider view of software engineering phenomena. They allow us to assess how widespread a phenomenon is, and thus, to focus future research on the most important issues. Tracy Hall, Sarah Beecham and Austen Rainer report a study of twelve companies that investigated the nature of requirement problems faced by those companies.

The last two papers in the Special Issue consider methodological issues directly. A common problem with empirical software engineering is that we often need to develop and validate special-purpose metrics. Manuel Serrano, Coral Calero and Mario Piattini present an example of how to validate metrics for data warehouses. Another problem is that software engineering data sets are quite small and it is difficult to obtain independent data sets to validate predictive models. Colin Kirsopp and Martin Shepperd investigate the extent to which the problem of small validation data sets undermines statistical inferences.

If this Special Issue has encouraged you to perform more empirical software engineering research, we encourage you to consider submitting a paper to or simply attending the next EASE conference in April 2003. For more information about the EASE conference checkout the EASE Web page at www.keele.ac.uk/depts/cs/ease.

BARBARA KITCHENHAM and DAVID BUDGEN

IEE Proceedings online no. 20020752
DOI: 10.1049/ip-opt:20020752

Barbara Kitchenham is Professor of Quantitative Software Engineering at Keele University. Her main research interest is software metrics and its application to project management, quality control, risk management and evaluation of software technologies. She is particularly interested in the limitations of technology and the practical problems associated with applying measurement technologies and experimental methods to software engineering. Kitchenham received a PhD from the University of Leeds. She is a Chartered Mathematician and Fellow of the Institute of Mathematics and Its Applications. She is also a Fellow of the Royal Statistical Society. She is a visiting professor at both the University of Bournemouth and the University of Ulster.

David Budgen is Professor of Software Engineering at Keele University, U.K. He received BSc and PhD degrees in theoretical physics from the University of Durham in 1969 and 1973. After a period working on naval command and control systems, where he developed an interest in software design issues, he was appointed to a post at the University of Stirling, moving to Keele University in 1991. His main research interests include software design, component based software engineering, CASE tools, and empirical software engineering.

Validating metrics for data warehouses

M. Serrano, C. Calero and M. Piattini

Abstract: Organisations are adopting data warehouses to manage information efficiently as the main organisational asset. This success of data warehouses (DW) can be explained because a data warehouse is a set of data and technologies aimed at enabling the executives, managers and analysts to make better and faster decisions. Due to the principal role of data warehouses in taking strategic decisions, quality is fundamental. One of the most important factors that affects the quality of the final system is its design. Although in recent years different authors have proposed some useful guidelines to design a data warehouse, more objective indicators are needed to help designers and managers to develop quality data warehouses. A set of metrics for data warehouse models is presented, and an empirical validation is carried out in order to prove practically their usefulness as quality indicators.

1 Introduction

It is known that organisations are very rich in data but poor in information. Today, technology has made it possible for organisations to store vast amounts of data obtained at a relatively low cost, although these data fail to provide information [1]. Data warehouses have appeared as a solution to this problem, supporting decision making processes and new kind of applications as marketing.

A data warehouse is defined as a 'collection of subject-oriented, integrated, non-volatile data that supports the management decision process' [2]. Data warehouses have become the key trend in corporate computing in recent years, since they provide managers with the most accurate and relevant information to improve strategic decisions. The future for data warehouses is also promising. Jarke *et al.* [3] forecast a market of twelve million US dollars for the data warehouse market over the next few years. However, the success of implementing a data warehouse and its use in the organisations could be seriously affected by the lack of quality.

Although some methodologies for data warehouse design have been proposed recently [4–6], they are not enough to assure data warehouse quality. Also, different authors have suggested interesting recommendations for achieving a 'good' dimensional data model [2, 6, 7]. However, from our point of view, the use of subjective quality criteria is not enough to ensure quality because different people could have different interpretations of the same concept. According to the total quality management (TQM) literature, measurable criteria for assessing quality are necessary to avoid 'arguments of style' [8]. So, objective mechanisms must be added to these methodologies and recommendations to assure the quality of the final product. This way, specific metrics, defined with a concrete goal, can be used to achieve objectively the desired quality.

Information quality can be decomposed into different types: presentation quality, data management system quality, data quality, physical model quality and multidimensional model quality. The last of these, and the definition of specific metrics for it, is our focus.

To summarise, the objective should be to replace intuitive notions of design 'quality' with formal, quantitative measures. In our case, these metrics will be defined for the multidimensional data model quality. In particular we will centre our work on defining metrics for its complexity, because complexity is believed to be an indicator of external quality attributes such as understandability, modifiability, etc. [9–12]. Although the empirical evidence supporting these relationships is scarce and suspect, in general, simpler schemas are easier to understand [13].

2 Defining valid metrics

Metric definition must be done in a methodological way, and it is necessary to follow a number of steps to ensure the reliability of the proposed metrics. Fig. 1 presents the method followed for the metric proposal [14].

In Fig. 1 we have three main activities:

- *Metrics definition:* The first step is the proposal of metrics. This definition is made taking into account the specific characteristics of the system we want to measure and the experience of designers of these systems. A goal-oriented approach, such as GQM (goal-question-metric) [15], can also be very useful in this step.
- *Theoretical validation:* The second step is the formal validation of the metrics. The formal validation helps us to know when and how to apply the metrics. There are two main tendencies in metrics validation: the frameworks based on axiomatic approaches [16, 17] and those based on measurement theory [12, 18, 19]. The strength of measurement theory is the formulation of empirical conditions from which we can derive hypotheses of reality. The final information when applying this kind of framework is to know to which scale a metric pertains, and based on this

---

IEE Proceedings online no. 20020697

DOI: 10.1049/ip-sen:20020697

Paper first received 19 April 2002 and in revised form 29 July 2002

The authors are with the Alarcos Research Group, Escuela Superior de Informática, University of Castilla—La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain

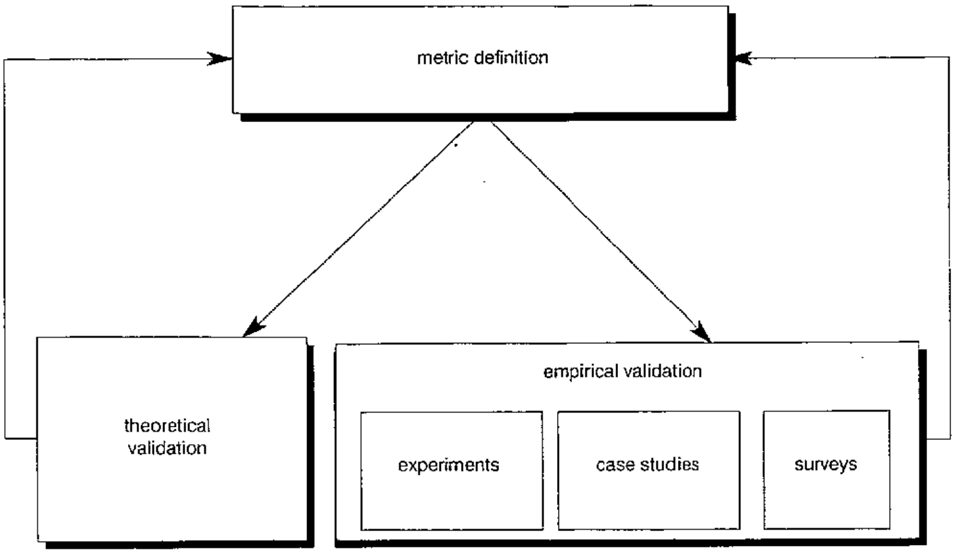

Fig. 1 *Steps followed in definition and validation of metrics*

Table 1: Metrics for data warehouse schemas

NFT(Sc). Number of fact tables of the schema.

NSDT(Sc). Number of shared dimension tables. Number of dimension tables shared for more than one star of the schema.

NAFT(Sc). Number of attributes of fact tables of the schema.

$$\mathrm{NAFT(Sc)} = \sum_{i=1}^{\mathrm{NFT}} \mathrm{NA(FT}_i)$$

Where FT_i is the fact table i of the schema Sc

NASDT(Sc). Number of attributes of shared dimension tables of the schema.

$$\mathrm{NASDT(Sc)} = \sum_{i=1}^{\mathrm{NSDT}} \mathrm{NA(DT}_i)$$

Where DT_i is the dimensional table i of the schema Sc

NFK(Sc). Number of foreign keys in all the fact tables of the schema.

$$\mathrm{NFK(Sc)} = \sum_{i=1}^{\mathrm{NFT}} \mathrm{NFK(FT}_i)$$

Where FT_i is the fact table i of the schema Sc

RT(Sc). Ratio of tables. Quantity of dimension tables per fact table.

$$\mathrm{RT(Sc)} = \frac{\mathrm{NDT(Sc)}}{\mathrm{NFT(Sc)}}$$

RFK(Sc). Ratio of foreign keys. Quantity of attributes that are foreign key.

$$\mathrm{RFK(Sc)} = \frac{\mathrm{NFK(Sc)}}{\mathrm{NA(Sc)}}$$

NDT(Sc). Number of dimension tables of the schema.

NT(Sc). Number of tables. Number of the fact tables plus the number of dimension tables of the schema.

$$\mathrm{NT(Sc)} = \mathrm{NFT(Sc)} + \mathrm{NDT(Sc)}$$

NADT(Sc). Number of attributes of dimension tables of the schema.

$$\mathrm{NADT(Sc)} = \sum_{i=1}^{\mathrm{NDT}} \mathrm{NA(DT}_i)$$

Where DT_i is the dimensional table i of the schema Sc

NA(Sc). Number of attributes of the schema.

$$\mathrm{NA(Sc)} = \mathrm{NAFT(Sc)} + \mathrm{NADT(Sc)}$$

RSDT(Sc). Ratio of shared dimension tables. Quantity of dimension tables, which belong to more than one star.

$$\mathrm{RSDT(Sc)} = \frac{\mathrm{NSDT(Sc)}}{\mathrm{NDT(Sc)}}$$

RScA(Sc). Ratio of schema attributes. Number of attributes in dimension tables per attributes in fact tables.

$$\mathrm{RScA(Sc)} = \frac{\mathrm{NADT(Sc)}}{\mathrm{NAFT(Sc)}}$$

RSDTA(Sc). Ratio of shared dimension tables Attributes. Number of attributes of the schema that are shared.

$$\mathrm{RSDTA(Sc)} = \frac{\mathrm{NASDT(Sc)}}{\mathrm{NA(Sc)}}$$

information we can know which statistics and which transformations can be done with the metric [18].

- *Empirical validation:* The goal of this step is to prove the practical utility of the proposed metrics. Although there are various ways of performing this step, basically we can divide the empirical validation into experiments, case studies and surveys [20–22].

As shown in Fig. 1, the process of defining and validating metrics is evolutionary and iterative. As a result of the feedback, metrics could be redefined or discarded based on theoretical or empirical validations.

3 Metrics for data warehouses

In this Section we present the metrics we have defined for data warehouses. Following the method of the previous Section, we will present first the metrics definition and afterwards we will made some comments about the other steps.

3.1 Metrics definition

Taking into account the characteristics of a data warehouse schema, we have defined the metrics shown in Table 1.

Fig. 2 shows an example of a data warehouse [7], and the values for the metrics can be found in Table 2. For example, the NA metric is calculated as the sum of the number of attributes of each of the tables in the schema (NA = NA(LOT/SERIAL) + NA(COMPONENT) + NA (DEFECT) + NA(SHIPMENT) + NA(FACILITY) + NA (SUPLIER) + NA(TIME) + NA(COMP_RECEIPT_FACTS) + NA(DEFECT_FACTS) = 4 + 6 + 4 + 3 + 3 + 5 + 6 + 7 + 8 = 46).

3.2 Theoretical validation

The metrics formal validation within the framework of [23] can be found in [24]. The results obtained are in Table 3.

3.3 Empirical validation

The next Section presents the controlled experiment we have carried out with the presented metrics, in order to know if they were useful as complexity mechanisms from a practical point of view.

4 Empirical validation of proposed measures

In this Section we describe a controlled experiment we have carried out to empirically validate the proposed metrics. We have followed some suggestions provided by [23, 25, 26] on how to perform controlled experiments. To describe the experiment we use (with only minor changes) we follow the format proposed in [23].

4.1 Definition

For the definition of the experimental goal, we can use the GQM template [15, 27]. In this way, the goal of the experiment can be defined as follows:

Analyse	*DW structural complexity measures*
For the purpose of	*evaluating*
With respect to	*the capability to be used as DW quality indicators*
From the point of view of	*database designers*
In the context of	*experts in database design*

4.2 Planning

(i) *Context selection:* The context of the experiment is a group of twelve experts in database design (practitioners with an average of two years on databases). The experiment is specific since it is focused on DW structural complexity measures (the ability to generalise from this specific context is further elaborated below when discussing threats to the study validity). The experiment addresses a real problem (i.e. what indicators can be used to assess

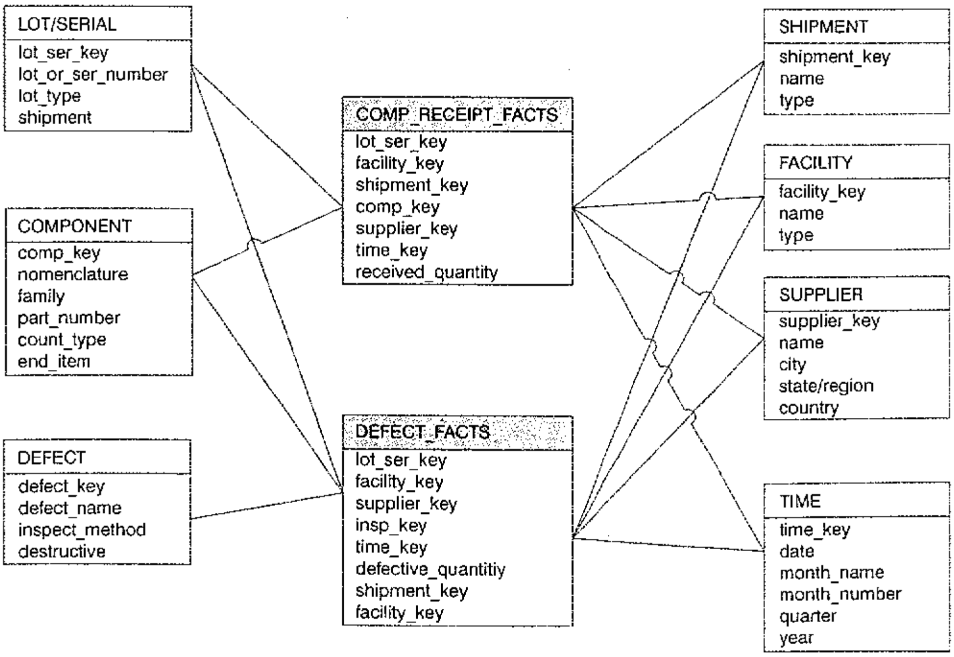

Fig. 2 *Example of a data warehouse star design [7]*

Table 2: Values for metrics

Metric	Value
NA	46
NFK	13
NDT	7
NT	9
NADT	31
NAFT	15
RFK	13/46
NFT	2
NSDT	6
NASDT	27
RSDT	6/7
RT	7/2
RScA	31/15
RSDTA	27/46

the quality of datawarehouses?) by investigating the correlation between DW structural complexity metrics and the understandability of datawarehouses (supposing that an understandable DW schema has more quality).
(ii) *Selection of subjects:* The subjects were chosen for convenience, i.e. the subjects were experts who had experience in database design, but without knowledge in datawarehouse design. We tried to select them with the same experience and background on databases.
(iii) *Variables selection:* The independent variable is DW structural complexity. The dependent variable is the understandability of the datawarehouse schema.
(iv) *Instrumentation:* The objects used in the experiment were eleven datawarehouse schemas. The independent variable was measured through the metrics we proposed in Section 3. The dependent variable was measured by asking each subject to rank the understandability of each schema (from 1 = too easy to 7 = too complex).
(v) *Hypotheses formulation:* We wish to test the following hypotheses:

(a) Null hypothesis, H_0: There is no a statistically significant correlation between the structural complexity measures and the understandability of the schemas.

Table 3: Scale for metrics

Metric	Scale
NA	Above ordinal
NFK	Above ordinal
NDT	Above ordinal
NT	Ratio
NADT	Above ordinal
NAFT	Above ordinal
RFK	Absolute
NFT	Ratio
NSDT	Above ordinal
NASDT	Ratio
RSDT	Absolute
RT	Absolute
RScA	Absolute
RSDTA	Absolute

(b) Alternative hypothesis, H_1: There is a statistically significant correlation between structural complexity measures and the understandability of the schemas that is also practically significant.

(vi) *Experiment design:* We selected a within-subject design experiment (i.e. all the tests had to be solved by each of the subjects) due to the low number of subjects. The subjects were given the schemas in different order.

4.3 Operation

(i) *Preparation:* Subjects were given an intensive explanation session before the experiment took place. However, subjects were not aware of the aspects we intended to study. Neither were they informed of the actual hypothesis stated.

We prepared the material we handed to the subjects, consisting of eleven DW schemas. These diagrams were related to different universes of discourse and were general enough to be easily understood by each of the subjects. The structural complexity of each diagram was different. In Table 4 the values of the metrics for each diagram are shown.

Each diagram had a test enclosed that asked the subjects to rank the complexity of the schema (from 1 = too easy to 7 = too complex).
(ii) *Execution:* The subjects were given all the materials described in the previous paragraph. We explained to them how to carry out the tests. We allowed one hour to do all the tests. Each subject had to work alone. In case of doubt, they could only consult the supervisor who organised the experiment.
(iii) *Data validation:* We collected all the tests, controlling whether they were complete and correct in order to know if some of them must be discarded. All the tests were correct and we were able to work with the twelve results for each schema.

4.4 Analysis and interpretation

We used the data collected in order to test the hypotheses formulated in Section 4.2. As we cannot assume that the data we collected follow a common statistical distribution (mainly because we have a very small group of subjects), we decided to apply a non-parametric correlational analysis, avoiding assumptions about the data normality.

In this way, we made a correlation statistical analysis using Kendall's tau_b, statistic which is a non-parametric test that avoids normality assumptions and is robust to the outliers. It can be used when there are duplicated values of the independent or dependent variables. For applying this test, the scale of the variables may be ordinal, ratio or interval. As we have duplicated values and each of the variables we have in the experiment is in a valid scale, we were able to use this test.

There is a threshold value for the correlation to be statistically significant. The threshold value depends on the sample size and the value of α selected for the analysis. In this way, the larger the correlation value the greater the statistical relationship between the two variables.

In our case, we used a level of significance $\alpha = 0.05$ (the experiment was a preliminary test and 0.05 is a common statistical value), which means a 95% level of confidence (i.e. the probability that we reject H_0 when H_0 is false is at least 95%, which is statistically acceptable). To calculate the cutoff value for accepting H_0, considering the selected α, we used the sample size (in our case twelve). The cutoff value for the correlation with $n = 12$ and

Table 4: Measure values for each DW schema

Schema	NFT	NDT	NSDT	NT	NAFT	NADT	NASDT	NA	NFK	RSDT	RT	RScA	RFK	RSDTA
1	1	4	0	5	5	19	0	24	4	0.00	4.00	3.80	0.17	0.00
2	3	7	2	10	19	39	11	58	11	0.29	2.33	2.05	0.19	0.19
3	3	5	4	8	15	28	22	43	10	0.80	1.67	1.87	0.23	0.51
4	4	8	4	12	23	42	24	65	17	0.50	2.00	1.83	0.26	0.37
5	1	3	0	4	4	14	0	18	3	0.00	3.00	3.50	0.17	0.00
6	1	5	0	6	16	39	0	55	5	0.00	5.00	2.44	0.09	0.00
7	2	5	4	7	18	32	25	50	10	0.80	2.50	1.78	0.20	0.50
8	1	4	0	5	10	25	0	35	4	0.00	4.00	2.50	0.11	0.00
9	1	3	0	4	11	25	0	36	3	0.00	3.00	2.27	0.08	0.00
10	1	6	0	7	7	46	0	53	6	0.00	6.00	6.57	0.11	0.00
11	2	5	4	7	12	28	23	40	9	0.80	2.50	2.33	0.23	0.58

$\alpha = 0.05$ is 0.5761 [28]. If the correlation value is greater than 0.5761 we can reject the null hypothesis (there is not significant correlation between the structural complexity measure and the understandability of the schemas), with a level of confidence of 95%.

Table 5 shows the results obtained, using the tau_b statistic, for the correlation between each of the measures and the complexity of the schemas.

Analysing Table 5, we can conclude that there is a high correlation between the complexity of the schemas and the metrics NFK, NFT, NT and NSDT (the correlation value is greater than 0.5761). The correlation value between the metric NDT and the complexity is less than the cutoff value. However, we cannot accept H_0 clearly due to the distance between the obtained value (0.571) and the cutoff value (0.576). The other metrics do not seem to be correlated with the complexity. In any case, it would be necessary to carry at more experimentation with the metrics to obtain more conclusive results about their usefulness as complexity indicators.

4.5 Validity evaluation

As we know, different threats to the validity of the results of an experiment exist. In this Section we are going to discuss threats to conclusion, construct, internal and external validity and the way we attempted to alleviate them.

Table 5: Tau_b correlation coefficients between measures and complexity of schemas

Tau_b correlation coefficient	Complexity
NA	0.448
NADT	0.365
NAFT	0.518
NASDT	0.520
NDT	0.571
NFK	0.664
NFT	0.688
NSDT	0.581
NT	0.669
RFK	0.520
RSCA	−0.406
RSDT	0.515
RSDTA	0.507
RT	−0.478

(i) *Threats to conclusion validity:* The conclusion validity defines the extent to which conclusions are statistically valid. One issue that could affect the statistical validity of this study is the size of the sample data (132 values, 11 diagrams and 12 subjects), which is perhaps not enough for both a parametric and non-parametric statistic test [26]. We are aware of this, so we will consider the results of the experiment only as preliminary findings.

(ii) *Threats to construct validity:* The construct validity is the degree to which the independent and dependent variables are accurately measured by the measurement instruments used in the study. Although the dependent variable reflects in some manner the complexity of the data warehouse schema, is it subjective, so we must consider the experiment as a first approach.

(iii) *Threats to internal validity:* The internal validity is the degree to which conclusions can be drawn about the causal effect of the independent variables on the dependent variables. The following issues have been dealt with:

(a) *Differences among subjects:* Using a within-subjects design, error variance due to differences among subjects is reduced.

(b) *Knowledge of the universe of discourse:* The DW schemas were from different universes of discourse but they are general enough to be easily understood for each of the subjects. We therefore believe that the knowledge of the domain does not considerably affect the internal validity.

(c) *Learning effects:* The subjects were given the tests in different order, to cancel out learning effects. Subjects were required and controlled to answer in the order in which the tests appeared.

(d) *Fatigue effects:* On average the experiment lasted for less than one hour, so fatigue was not very relevant. Also, the different order in the tests helped to cancel out these effects.

(e) *Persistence effects:* In order to avoid persistence effects, the experiment was run with subjects who had never done a similar experiment.

(f) *Subject motivation:* We motivated subjects to participate in the experiment, explaining to them that the results of the experiment could benefit them as information systems practitioners.

(g) *Other factors:* Subjects were told to not talk to one another.

(iv) *Threats to external validity:* The external validity is the degree to which the results of the research can be generalised to the population under study and to other research settings. The greater the external validity, the more the results of an empirical study can be generalised to actual software engineering practice. Two threats to validity have been identified which limit the ability to apply any such generalisation:

(a) *Materials and tasks used:* In the experiment we tried to use DW schemas which can be representative of real cases, but more empirical studies using 'real cases' from software companies must be carried out.
(b) *Subjects:* We are aware that more experiments with practitioners and professionals must be carried out in order to be able to generalise these results.

5 Conclusions and future work

If we really consider that information is the main organisational asset, one of our primary duties should be to assure its quality. Although some interesting guidelines have been proposed for designing 'good' multidimensional models, more objective indicators are needed. Metrics are useful and objective mechanisms for improving the quality of software products, and also for determining the best ways to help professionals and researchers.

In this way, our goal is to elaborate a set of metrics for measuring data warehouse quality which can help designers in choosing the best option from alternative designs.

To this end, we have presented in this paper some metrics defined for measuring the data warehouse star design complexity, and we have explained a first experiment that has been developed in order to validate the metrics.

Similar empirical research investigating the relationships between the internal quality attributes of several software systems and its understandability have been published in [29–31]. To our knowledge, there are no related works in the area of data warehouse models.

As a conclusion of our experiment, there seems to be a correlation between some of the metrics (NT, NFT, NSDT, and maybe NDT) and the complexity of the data warehouse schema. Even though the results obtained in this experiment are encouraging we cannot consider them as conclusive results. We are aware that it is necessary to replicate the experiment and to carry out new ones in order to confirm our results. It is also necessary to undertake case studies working with real data.

6 Acknowledgments

This research is part of the CALDEA project (TIC 2000-0024-P4-02) supported by the Ministerio de Ciencia y Tecnologia of Spain.

7 References

1 GARDNER, S.R.: 'Building the data warehouse', *Commun. ACM*, September 1998, **41**, (9), pp. 52–60

2 INMON, W.H.: 'Building the data warehouse' (John Wiley and Sons, New York, 1997), 2nd edn

3 JARKE, M., LENZERINI, M., VASSILOU, Y., and VASSILIADIS, P.: 'Fundamentals of data warehouses' (Ed. Springer, 2000)

4 HAMMERGREN, T.: 'Data warehousing building the corporate knowledge base' (International Thomson Computer Press, Milford, 1996)

5 KELLY, S.: 'Data warehousing in action' (John Wiley & Sons, New York, 1997)

6 KIMBALL, R., REEVES, L., ROSS, M., and THORNTHWAITE, W.: 'The data warehouse lifecycle toolkit' (John Wiley and Sons, New York, 1998)

7 ADAMSON, C., and VENERABLE, M.: 'Data warehouse design solutions' (John Wiley and Sons, 1998)

8 BOMAN, M., BUBENKO, J., JOHANNESSON, P., and WANGLER, B.: 'Conceptual modelling' (Prentice Hall, 1997)

9 ZUSE, H.: 'Encyclopedia of software engineering' (John Wiley and Sons, New York, 1994)

10 HENDERSON-SELLERS, B.: 'Object-oriented metrics—measures of complexity' (Prentice-Hall, New Jersey, 1996)

11 LI, H.F., and CHENG, W.K.: 'An empirical study of software metrics', *IEEE Trans. Softw. Eng.*, 1987, **13**, (6), pp. 679–708

12 WHITMIRE, S.A.: 'Object oriented design measurement' (Ed. Wiley, New York, 1997)

13 MOODY, L., and SHANKS, G.: 'What makes a good data model? Evaluating the quality of entity relationships models'. 13th International Conference on Conceptual modelling, E/R'94, December 1994, Manchester, England, pp. 94–111

14 CALERO, C., PIATTINI, M., and GENERO, M.: 'Empirical validation of referential integrity metrics', *Inf. Softw. Technol.*, 2001, **43**, (15), pp. 949–957

15 BASILI, V.R., and WEISS, D.: 'A methodology for collecting valid software engineering data', *IEEE Trans. Softw. Eng.*, 1984, **10**, (6), pp. 728–738

16 WEYUKER, E.J.: 'Evaluating software complexity measures', *IEEE Trans. Softw. Eng.*, 1988, **14**, (9), pp. 1357–1365

17 BRIAND, L.C., MORASCA, S., and BASILI, V.: 'Property-based software engineering measurement', *IEEE Trans. Softw. Eng.*, 1996, **22**, (1), pp. 68–85

18 ZUSE, H.: 'A framework of software measurement' (Walter de Gruyter, Berlin, 1998)

19 POELS, G., and DEDENE, G.: 'Distance-based software measurement: necessary and sufficient properties for software measures', *Inf. Softw. Technol.*, 2000, **42**, (1), pp. 35–46

20 BASILI, V.R., SHULL, F., and LANUBILLE, F.: 'Building knowledge through families of experiments', *IEEE Trans. Softw. Eng.*, 1999, **25**, (4), pp. 456–473

21 FENTON, N., and PFLEEGER, S.: 'Software metrics: a rigorous aproach' (Chapman & Hall, London, 1997, 2nd edn.)

22 PFLEEGER, S.A., and KITCHENHAM, B.A.: 'Principles of survey research', *Softw. Eng. Notes*, 2001, **26**, (6), pp. 16–18

23 WOHLIN, C., RUNESON, P., HÖST, M., OHLSON, M., REGNELL, B., and WESSLÉN, A.: 'Experimentation in software engineering: an introduction' (Kluwer Academic Publishers, Boston, 2000)

24 CALERO, C., PIATTINI, M., PASCUAL, C., and SERRANO, M.: 'Towards datawarehouse quality metrics'. International Workshop on Design and management of data warehouses, DMDW'01 in conjuntion with the 13th Conference on Advanced information systems engineering, CAISE'01, 2001

25 PERRY, D., PORTER, A., and VOTTA, L.: 'Empirical studies of software engineering: A roadmap. Future of Software Engineering' (Ed:Anthony Finkelstein, ACM, 2000, pp. 345–355

26 BRIAND, L., EL EMAM, K., and MORASCA, S.: 'Theoretical and empirical validation of software product measures'. Technical report ISERN-95-03, International Software Engineering Research Network, 1995

27 BASILI, V., and ROMBACH, H.: 'The TAME project: towards improvement-oriented software environments', *IEEE Trans. Softw. Eng.*, 1988, **14**, (6), pp. 728–738

28 DEPARTMENT OF OBSTRETICS AND GYNAECOLOGY—THE CHINESE UNIVERSITY OF HONK KONG: Available at http://department.obg.cuhk.edu.hk/ResearchSupport/Minimum_correlation.asp (accessed on July 22nd, 2002).

29 HARRISON, R., COUNSELL, S., and NITHI, R.: 'Experimental assessment of the effect of inheritance on the maintainability of object-oriented systems', *J. Syst. Softw.*, 2000, **52**, pp. 173–179

30 BRIAND, L., BUNSE, C., and DALY, J.: 'A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs', *IEEE Trans. Softw. Eng.*, 2001, **27**, (6), pp. 523–530

31 GENERO, M., POELS, G., and PIATTINI, M.: 'Defining and validating measures for conceptual data model quality'. 14th International Conference on Advanced information systems engineering, CAISE 2002, 27–31 May 2002, Toronto, Canada, pp. 724–727

CALL FOR PAPERS

IEE Proceedings-Software

Special Issue on Information Retrieval and Personalisation on the Internet

This special issue will consist of invited and contributed papers discussing the current frontiers in Information Retrieval and Personalisation on the Internet.

The Internet has opened up new ways of providing access to information - with new problems and new challenges. The biggest and most obvious of these, that of finding relevant information in response to a query, has attracted much attention. Although much effort has been devoted to the development of search engines and there has been some improvement in the techniques used, there is still much work to be done in this area. The idea of developing and maintaining a user profile for each individual user and using this to perform personalised search to produce the most relevant results for any particular situation is still a distant dream.

Providing information via the Internet also opens up new opportunities in the way information is presented to the user. In particular, the notions of personalisation are attracting growing interest. The basic idea is that the system may respond differently to different users issuing the same query, selecting different content, ordering it differently or presenting it in different ways according to the user's preferences and interests. Where there is insufficient information on the user, the system may infer possible interests on the basis of known interests of similar users. The techniques are used in eCommerce as well as in other application areas, e.g. healthcare.

The aim of this special issue is to bring together current research on different aspects of personalisation and information retrieval on the Internet to provide a picture of the current state of technology. Contributions dealing with theory, techniques, and performance evaluation, are solicited, but not limited to, the following:

- Personalisation techniques
- Intelligent information retrieval
- User profile acquisition and maintenance
- Performance evaluation

Potential contributors may contact the guest editor to discuss the potential suitability of their contribution:

Prof. Howard Williams
Professor of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University
Riccarton
Edinburgh
EH14 4AS

+44 131 451 3430 (phone)
+44 131 451 3327 (fax)
mhw@cee.hw.ac.uk (email)

No limit is imposed on the length of submitted manuscripts. All submitted papers will be subject to the IEE Proceedingsí stringent refereeing procedures. Further guidance for authors can be obtained at http://www.iee.org/Publish/Journals/Profjourn/Proc/sen/#Author_guide. First manuscripts should be submitted in PDF form to the editorial assistant, Mr Lee Baldwin, at lbaldwin@iee.org.uk and should be accompanied by a covering email stating clearly the title of the special issue to which they are submitted. The deadline for the submission of first manuscripts is 1 May 2003

Guide to authors of papers for IEE Proceedings—Software

A more detailed guide to authors can be found at http://www.iee.org/Publish/Journals/Profjourn/Proc/sen/#Author_guide

Scope

IEE Proceedings-Software aims to publish papers on all aspects of the software lifecycle, including design, development, implementation and maintenance. The focus of the journal is on the methods used to develop and maintain software, and their practical application. Authors are especially encouraged to submit papers on the following topics, although papers on all aspects of software engineering are welcome:

- software and systems requirements engineering
- formal methods, design methods, practice and experience
- software architecture, object orientation, reuse and re-engineering
- testing, verification and validation techniques
- software dependability software measurement
- human systems engineering: human–computer interaction
- knowledge engineering: expert and knowledge-based systems, intelligent agents
- information systems engineering
- application of software engineering in industry and commerce
- software engineering technology transfer
- management of software development
- theoretical aspects of software development
- education and training including continuing professional development

Peer review

All papers are rigorously refereed. Multiple submissions are strongly discouraged and action may be taken against authors who have submitted the same work to other journals. It is our policy to ensure that a decision to publish or not is made and the authors notified within six months.

Submission

Authors are requested to email their entire manuscript for referral purposes as a single PDF document. Submissions should be sent to proceedings@iee.org.uk. Authors of accepted papers will be contacted to supply their final manuscript in an electronic format suitable for production. PDF proofs of typeset papers will be sent to authors for final approval. Corrections to these proofs should be returned to the editorial office within 72 hours of receipt.

Language

To ensure the widest possible readership, papers must be written in English.

Length

There is no strict limit on length, although manuscripts should generally not exceed 10–20 double-spaced A4 pages, plus tables and illustrations.

Typescript

The complete typescript, i.e. abstract, text of the paper, references and figure captions, should be typed with double line spacing.

Affiliations of authors

The affiliation and full postal address of each author should be given.

Abstracts

Each paper should be accompanied by a self-contained abstract suitable for publication and indexing, of no more than 250 words.

Photographs and illustrations

Illustrations enclosed when a paper is first submitted should be suitable for reproduction. Colour should only be used where this is absolutely necessary for an understanding of the figure and not, for example, to distinguish line data.

Each figure should be drawn on a separate page at the end of the manuscript.

The authors should obtain from the owners of the copyright written permission to reproduce any illustration for which the copyright is not their own. The source of the illustration must be given in full and the words 'Reproduced by permission of . . .' included with the illustration where necessary.

References

Other publications referred to in the text should be indicated by a number and should be cited in ascending numerical order. Details of the references should be given in a list at the end of the paper in order of citation and should include names of all authors, title of the paper, full title of the journal, year of publication and volume number, and first and last page numbers. For a book, the author, book title, publisher and year of publication should be stated, while for web-based references, the author, title and URL should be given.

PROFESSOR JUDITH BISHOP